Generalised Zero-Shot Learning via Novelty Detection

Student Name: D. Hopkinson-Sibley

Supervisor Name: Y. Long

Submitted as part of the degree of BSc Computer Science to the

Board of Examiners in the Department of Computer Sciences, Durham University

***Abstract* —**

Context/Background: Large, labeled datasets are required to achieve a high level of accuracy with traditional approaches to image classification. This poses a significant challenge when sources of data for certain classes we would like to train our models on are scarce or unavailable. Zero-shot learning, and its extension, generalised zero-shot learning, offer a solution to this problem.

Aims: To develop a new framework for generalised zero-shot learning which performs as good as or better than the current state-of-the-art.

Method: An existing ZSL model is augmented with a novelty detection model in a novel approach which estimates the probability of a test sample belonging to a class that was not present during training. This information aids the ZSL model in choosing correctly between the set of seen or unseen classes. In addition, the ZSL model bias is measured in a novel way and incorporated into the framework to increase overall accuracy.

Results: The proposed framework is tested on four benchmark datasets and achieves near state-of-the-art accuracy on two of them, showing significant improvement over baseline methods and other novelty-detection-based approaches across all datasets.

Conclusions: In addition to the proposed framework, a generic way in which zero-shot learning and novelty detection models can be combined is demonstrated, showing highly promising results compared to current methods.

***Keywords* —** Generalised Zero-Shot Learning, Novelty Detection, Image Classification

I INTRODUCTION

While deep convolutional neural networks such as ResNet (He et al. 2016) and more recently SENet (Hu et al. 2018) have surpassed human-level performance on the ImageNet Large Scale Visual Recognition Challenge, these methods require thousands of labelled training examples in order to generalise well to the test set. In real-world applications, obtaining such a large amount of training data for each class of object can be impractical. Zero-shot learning (ZSL) has gained much attention recently from the AI research community because it aims to overcome this problem. Usually there is a certain amount of overlap between different classes of images. For instance, images of horses and zebras share a lot of common structures in the body shape, but differ in terms of the colours and patterns of the skin. ZSL aims classify images of classes

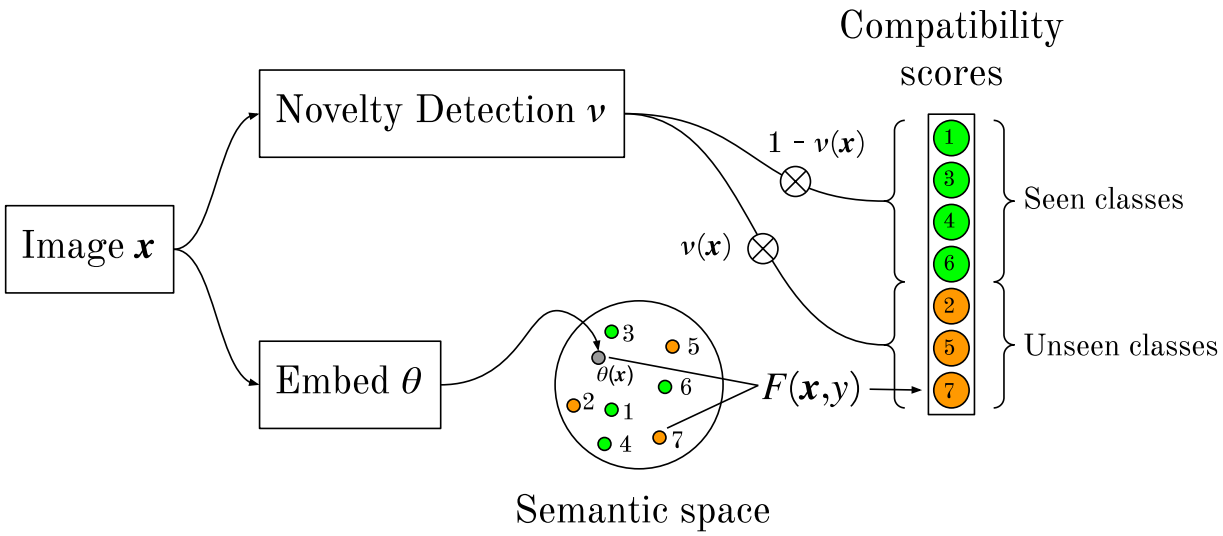

Figure 1: Diagram of proposed GZSL framework. The output from the novelty detection model, $\nu(\boldsymbol{x})$, aids in discriminating between seen and unseen classes for zero-shot learning by augmenting the compatability scores.

for which it has no training data by exploiting these relationships between classes. The most common approach, first popularised by the seminal work of Lampert et al., is to use a set of attributes for each class. This effectively allows a mapping from the image space to the attribute space to be learned, meaning the attributes of an image can be predicted and thus a class label predicted, even though no training examples of this class are ever seen by the classifier.

Clearly, zero-shot learning is of practical importance as well as theoretical. The large-scale datasets required by today's deep networks are typically very expensive and laborious to gather, process and annotate. Services like Amazon's Mechanical Turk have helped researchers and practitioners to outsource the task of image annotation to the general public. Sometimes this is not an option, for example in the case of fine-grained or domain-specific datasets, expert knowledge may be required to label the data. Moreover, new classes may emerge in the future which we would still like our models to be able to classify correctly without gather a new set of data for these categories. Zero-shot learning will help to overcome these practical problems if it is feasible to provide prior semantic knowledge to the model about the unseen classes.

While much work has been done in tackling the problem of ZSL, recent attention has turned towards generalised zero-shot learning (GZSL). This setting describes the more realistic scenario where test images may belong to either seen or unseen classes and so all classes must be considered by the model. Almost all traditional ZSL frameworks exhibit strong bias towards the training classes which arises due to the fact that a particular attribute may manifest itself differently between classes. As a result, the image-attribute projection learned is inherently biased towards the seen classes. This was first described by Fu et al. as the *domain shift problem*.

In this project, I aim to address this problem in GZSL by augmenting the ZSL model with information from a novelty detection model. The idea is that, by knowing how likely an image at test time is to belong to a training class or not, we can 'push' the ZSL model to favour choosing a class from either the seen or unseen set. I will show that this addition leads to better GZSL

test accuracy overall. The research question I am addressing can be stated as, "how can novelty detection be incorporated into zero-shot learning so as to improve the performance in the generalised setting." Further, the bias of the ZSL model in favor of predicting seen/unseen classes is investigated, with a novel metric of this bias proposed. This metric, measured on a validation set, can then be incorporated into the framework to give further improvements to the accuracy, albeit sometimes with a trade-off between accuracy on the seen and unseen classes.

II RELATED WORK

A *Zero-shot learning*

The formal definition of zero-shot learning is as follows. Let $\mathcal{X}$ be a feature space and $\mathcal{Y}^s$ and $\mathcal{Y}^u$ be sets of object classes where $\mathcal{Y}^s \cap \mathcal{Y}^u = \emptyset$. The task is to learn a classifier $f : \mathcal{X} \rightarrow \mathcal{Y}^u$ by using only training examples $\{(\boldsymbol{x}^{(0)}, y^{(0)}), ..., (\boldsymbol{x}^{(n)}, y^{(n)})\} \subset \mathcal{X} \times \mathcal{Y}^s$. One of the first works on zero-shot image classification was that of direct attribute prediction (DAP) and indirect attribute prediction (IAP) (Lampert et al. 2014). These methods rely on the existence of an attribute vector $a_m \in \mathcal{A}$ for every class. In DAP, a probabilistic classifier $p(a^m|\boldsymbol{x})$ is learned for each attribute m. For example, given the features of a polar bear image, it is trained to output high probability for the attribute *eats fish* and low probability for *domestic*. During test time we classify an unseen image as one of the unseen classes $\mathcal{Y}^u$ such that the probability of this image having that class' attributes is maximum. In IAP, the attribute probabilities are estimated indirectly by training a standard multiclass classifier $p(c|\boldsymbol{x})$ for each seen class $c \in \mathcal{Y}^s$. During test time, the attribute probability $p(a^m|\boldsymbol{x})$ is simply the sum of the class probabilities weighted by the attribute magnitude, and the predicted class is obtained the same way as in DAP.

Attribute-Label Embedding (ALE) (Akata et al. 2016) improved upon IAP/DAP in many ways. Rather than learn independent classifiers for each attribute, it learns an embedding function $\theta : \mathcal{X} \rightarrow \mathcal{A}$ from the image features to the class attributes, known as the *semantic space* more generally. This removes the attribute independence assumption in DAP/IAP. In addition, it proposed the idea of using *label embedding*, a function $\varphi : \mathcal{Y} \rightarrow \mathcal{A}$ which maps classes to attributes. This allows other sources of side information to be used, such as Word2Vec word embeddings (Mikolov et al. 2013). The compatibility function $F(\boldsymbol{x}, y)$ calculates the similarity between $\theta(\boldsymbol{x})$ and $\varphi(y)$ to enable class predictions.

B *Generalised zero-shot learning*

Generalised zero-shot learning describes the setting where test images may belong to seen or unseen classes, therefore the classifier to be learned is of the form $f : \mathcal{X} \rightarrow \mathcal{Y}$ where $\mathcal{Y} = \mathcal{Y}^u \cup \mathcal{Y}^s$. The difficulty in GZSL was demonstrated by (Chao et al. 2016) where they analysed the performance of existing ZSL techniques and showed a clear bias towards predicting seen classes. Using Iverson's bracket notation where $[\![P]\!]$ is 1 if P is true and 0 if P is false, their proposed simple modification to the compatibility function is

$$F_c(\boldsymbol{x}, y) = F(\boldsymbol{x}, y) - \gamma[\![y \in \mathcal{Y}^s]\!], \tag{1}$$

where γ is a hyperparameter. This helped reduce the rate of misclassifications of unseen classes, although in turn reduced the performance on the seen classes. Recently, generative models such as UVDS (Long et al. 2017) and f-CLSWGAN (Xian et al. 2018) have shown superior results in

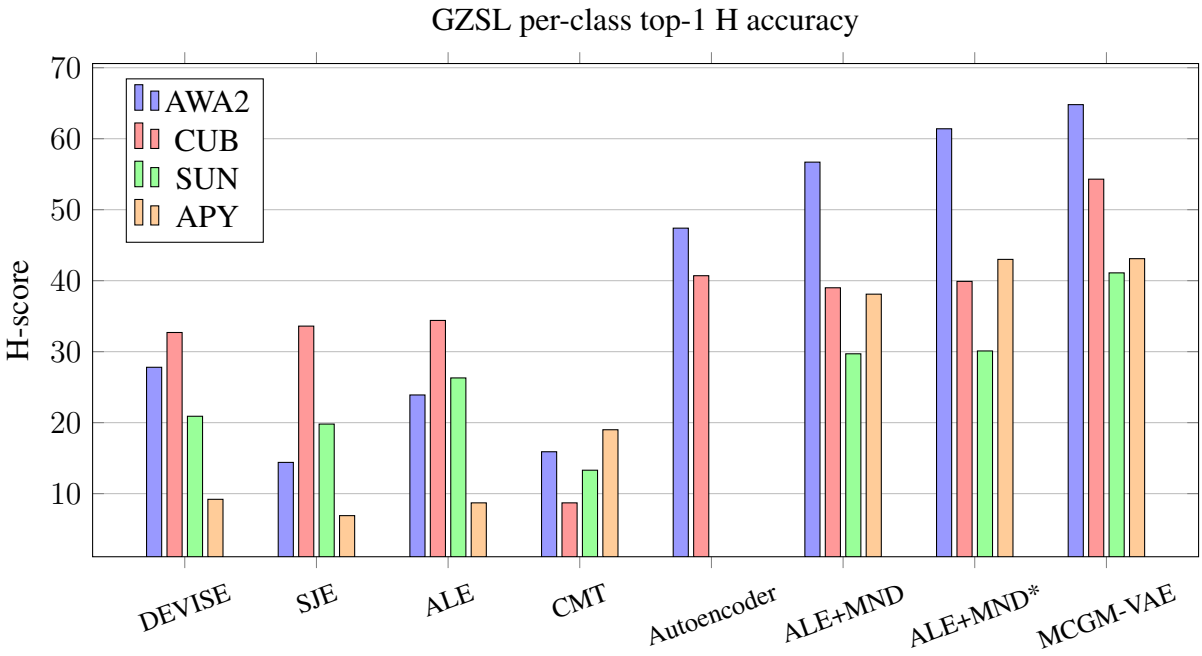

Figure 2: Comparison of GZSL per-class top-1 accuracy of proposed approach (ALE+MND) vs. existing methods (Autoencoder = SAE + autoencoder novelty detection (Bhattacharjee et al. 2019), ALE+MND = $\nu_{Gaussian}$, ALE+MND* = $\nu_{Gaussian}$ with de-biasing).

ZSL by generating synthetic image features for unseen classes, reducing the task of ZSL to conventional image classification. The advantage of generative approaches is that they automatically reduce bias towards seen classes by generating an equal number of synthetic samples for unseen classes. UVDS learns a linear mapping from attributes to image features while f-CLSWGAN trains an attribute-conditional Generative Adversarial Network (Goodfellow et al. 2014) to generate image features. It was shown that training a standard softmax classifier trained on the set of real and synthetic features could outperform all other techniques at the time. Generative approaches to GZSL are currently the state of the art (Shao & Li 2020).

C Novelty detection

Novelty detection has previously been used to tackle the GZSL problem by selecting a traditional softmax classifier if the probability of a sample belonging to a seen class is high, and ZSL model if it is low (Socher et al. 2013). They use a simple outlier detection scheme whereby the embedding of each class prototype in the semantic space is modelled as a multivariate Gaussian and the probability of observing a new sample $\theta(\boldsymbol{x})$ from one of the seen classes is then thresholded. This technique did not achieve good results, however, and in some cases even *reduced* GZSL performance (particularly for fine-grained datasets where novelty detection is more difficult). This is likely to the over-simplistic determination of outlier classes. The most successful attempt at improving ZSL model performance in the generalised setting is that of Bhattacharjee et al. (2019). They use an autoencoder which takes as input a sample $\boldsymbol{x}$ and attribute vector a and reconstructs x. The maximum cosine similarity between a test sample and its reconstruction for each seen class' attribute vector is thresholded to obtain the novelty. Novelty detection has also

been used for the task of generalised zero-shot action recognition, the task of classifying human actions/movements from video footage (Mandal et al. 2019). Interestingly, they use a combination of the generative and novelty detection approaches, by training their out-of-distribution detection model on both real and synthesised image features with a GAN.

D Proposed approach

This work is different to the existing GZSL approaches using novelty detection in a few ways. First, rather simply thresholding the novelty probability and using this to choose between the set of seen or unseen classes, the novelty information is incorporated into the ZSL model in a probabilistic fashion, by weighting the compatibility scores appropriately. Second, the novelty detection mechanisms used by existing methods are specific to the task of GZSL, meaning this part of the model cannot easily be swapped for another solution. In contrast, the proposed approach uses a novelty detection model which needs only the image features of seen classes to be trained, and outputs an arbitrary value that can be separately converted into a novelty probability. This means the novelty detection component can swapped out for a better performing alternative with no changes to the rest of the framework. Finally, none of the existing works has directly measured the bias of the ZSL model beyond the classification accuracy. In this work a more descriptive bias metric is proposed and incorporated into the framework to reduce the overall bias and increase accuracy.

III SOLUTION

A Overview

The proposed solution uses a 2-stage process to predict the correct class $y^{(i)} \in \mathcal{Y}$ of a sample $\boldsymbol{x^{(i)}} \in \mathcal{X}$ where $\mathcal{X}$ is the image feature space. In the first stage, a novelty detection framework takes as input $x^{(i)}$ and outputs a membership score $N(\boldsymbol{x}) \in \mathbb{R}$. This model is trained with the expectation that, for samples belonging to the set of seen classes ($y^{(i)} \in \mathcal{Y}^s$), a high score will be generated on average, while for unseen classes ($y^{(i)} \in \mathcal{Y}^u$) the score will be lower. A function $\nu : \mathbb{R} \rightarrow [0, 1]$ is then applied to the membership score to give a probability of novelty. This novelty function can take many different forms, but I have evaluated four different types, which I will call *thresholding*, *logistic regression*, *percentiles* and *Gaussian*. In the second stage, $\boldsymbol{x^{(i)}}$ is passed through a zero-shot learning model which learns a compatibility function $F : \mathcal{X} \times \mathcal{Y} \rightarrow \mathbb{R}$. The output of this function $F(\boldsymbol{x^{(i)}}, y)$ is the compatibility between $\boldsymbol{x^{(i)}}$ and the class y. Without consideration of the novelty information, the predicted class is simply

$$\hat{y}^{(i)} = \underset{y \in \boldsymbol{Y}}{\arg\max}\, F(\boldsymbol{x^{(i)}}, y) \tag{2}$$

. The novelty information is incorporated into the ZSL model by weighting the compatibility of each class differently for seen and unseen classes. The predicted class is now given by

$$\hat{y}^{(i)} = \underset{y \in \boldsymbol{Y}}{\arg\max} \begin{cases} \nu(\boldsymbol{x^{(i)}}) F(\boldsymbol{x^{(i)}}, y) & \text{if } y \in \mathcal{Y}^u \\ (1 - \nu(\boldsymbol{x^{(i)}})) F(\boldsymbol{x^{(i)}}, y) & \text{otherwise.} \end{cases} \tag{3}$$

Figure 1 shows an overview of the framework.

B Novelty functions

B.1 Thresholding

The first and most simple choice for the novelty function is to threshold the membership score as follows:

$$\nu_{threshold}(\boldsymbol{x}) = [\![N(\boldsymbol{x}) > t]\!] \tag{4}$$

again using Iverson's bracket notation, where t is obtained through cross-validation so as to maximise the TPR $-$ FPR (true positive rate, false positive rate).

B.2 Logistic regression

It might be desirable to have a confidence on whether or not a sample belongs to one of the novel classes, instead of just the true or false value given by thresholding. For example, if the membership score of a novel sample just above the threshold, it would incorrectly classified by the ZSL model even if the compatibility of the correct class was highest. Having a continuous value for the novelty will become particularly useful when tackling the problem of to de-biasing below. One way of achieving this is by using the following equation

$$\nu_{logistic}(\boldsymbol{x}) = \sigma(aN(\boldsymbol{x}) + b) \tag{5}$$

where σ is the logistic sigmoid function and a and b are constants found through cross validation. This function can be optimised through logistic regression on the membership scores and binary labels from the validation set by minimising the negative log-likelihood function:

$$J(a,b) = -\sum_{i=1}^{N} \left[[\![y^{(i)} \in \mathcal{Y}^u]\!] \log(\nu(\boldsymbol{x}^{(i)}; a, b)) + [\![y^{(i)} \in \mathcal{Y}^s]\!] \log(1 - \nu(\boldsymbol{x}^{(i)}; a, b)) \right]. \tag{6}$$

B.3 Using score distributions

Exploiting the fact that we have a large number of membership scores for both seen and unseen classes from the validation set, we can compare the score of a new sample to both of these distributions to produce a probability. If F_S is the cumulative distribution function of the seen scores, and F_U that of the unseen scores, then the unnormalised probability of a sample belonging to a seen class is $F_S(N(\boldsymbol{x}))$ and the probability of being novel is $1 - F_U(N(\boldsymbol{x}))$. The normalised novelty probability is given by:

$$\nu_{cdf}(\boldsymbol{x}) = \frac{1 - F_U(N(\boldsymbol{x}))}{1 - F_U(N(\boldsymbol{x})) + F_S(N(\boldsymbol{x}))}. \tag{7}$$

These CDFs can be estimated from the validation sets.

B.4 Gaussian

We can simplify the above method by modelling the distributions of scores as Gaussians. The mean and variance are estimated from the membership scores of the validation sets, and the function is the same as (7) but with Gaussian CDFs used. This will be called $\nu_{Gaussian}$.

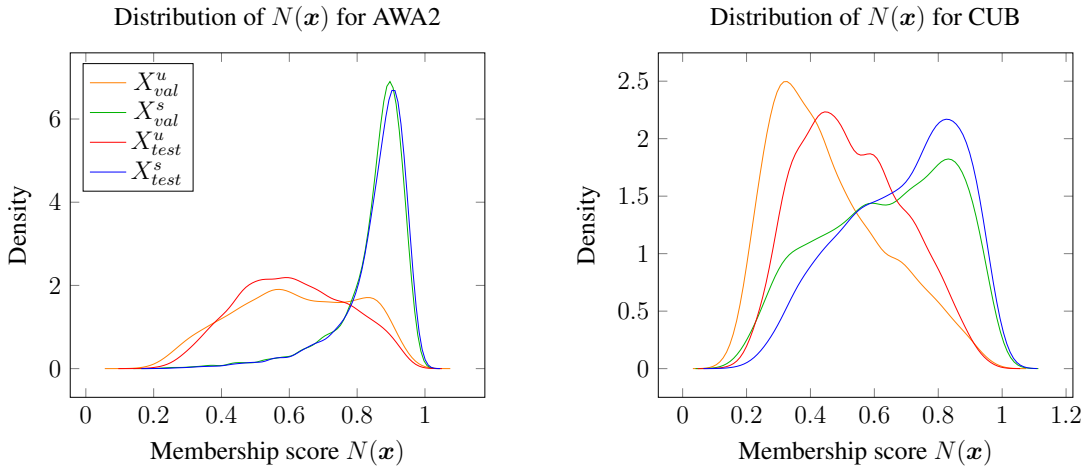

Figure 3: Distribution of membership score $N(\boldsymbol{x})$ for AWA2 and CUB validation and test splits, smoothed with kernel density estimation. Distributions are more similar for AWA2 than for CUB, and the seen/unseen distributions overlap less.

C *De-biasing*

The *domain shift* problem identified by Fu et al. (2015) describes the tendency of most zero-shot learning frameworks to exhibit a bias towards the seen classes at test time. This is the major difficulty in generalised zero-shot learning, since both seen and unseen classes must be considered. This can be seen by the difference in accuracy of ALE between $\mathcal{Y}^s$ and $\mathcal{Y}^u$ in table 4. In order to help overcome the bias of the ZSL model, I propose to add an equal and opposite bias to the novelty function ν. This requires a formal definition of the bias of the compatibility function F of a ZSL model. One possible definition could be the likelihood ratio between a model predicting some label $s \in \mathcal{Y}^s$ and some other label $u \in \mathcal{Y}^u$ for a random test sample $\boldsymbol{x}$. While this is a valid definition, it does not help much in seeing the reason for the bias. Instead, I choose to define the bias as the expected ratio between the highest compatibility score of all seen classes and the highest score of all the unseen classes:

$$bias = \mathbb{E}_{\boldsymbol{x} \sim p_{data}} \left[\frac{\max_{s \in \mathcal{Y}^s} F(\boldsymbol{x}, s)}{\max_{u \in \mathcal{Y}^u} F(\boldsymbol{x}, u)} \right] \tag{8}$$

This can be estimated for some validation set X as follows:

$$bias(X) = \sum_{\boldsymbol{x} \in X} \frac{\max_{s \in \mathcal{Y}^s} F(\boldsymbol{x}, s)}{\max_{u \in \mathcal{Y}^u} F(\boldsymbol{x}, u)} \tag{9}$$

If the validation set itself has a large difference in the number of samples from seen and unseen classes, this would cause the bias estimate to be biased itself. Therefore, the overall bias is estimated as:

$$\hat{b} = \frac{1}{2} \left(bias(X^u_{val}) + bias(X^s_{val}) \right) \tag{10}$$

Table 1: BIAS OF ALE MEASURED ON VALIDATION SETS

Split	AWA2	CUB	SUN	APY
1	6.17	1.14	1.07	22.97
2	6.51	1.10	1.07	9.22
3	3.71	1.11	1.07	3.81
Mean	**5.46**	**1.12**	**1.07**	**12.00**

Table 1 shows the average bias across all validation splits for each dataset. In order to counter this bias I modify the novelty function ν by skewing it towards 1. More specifically, for logistic regression, the class 'novel' is weighted by $\hat{b}$, meaning mistakes in misclassifying a novel sample are penalised $\hat{b}$ times more than seen samples. To loss function to be minimised becomes

$$J^*(\theta, \hat{b}) = -\sum_{i=1}^{N} \left[[\![y^{(i)} \in \mathcal{Y}^u]\!] \hat{b} \log(\nu(\boldsymbol{x}^{(i)}; \theta)) + [\![y^{(i)} \in \mathcal{Y}^s]\!] \log(1 - \nu(\boldsymbol{x}^{(i)}; \theta)) \right]. \tag{11}$$

I'll denote this weighted version $\nu^*_{logistic}$. For ν_{cdf}, the probability of a sample being novel is weighted by $\hat{b}$, so the equation becomes

$$\nu^*_{cdf}(\boldsymbol{x}, \hat{b}) = \frac{\hat{b}(1 - F_U(N(\boldsymbol{x})))}{\hat{b}(1 - F_U(N(\boldsymbol{x}))) + F_S(N(\boldsymbol{x}))}. \tag{12}$$

The same modification is made in $\nu^*_{Gaussian}$. $\nu_{threshold}$ gives a binary value rather than a probability, so it does not make sense to skew it without obviously negatively impacting the accuracy on seen classes.

D Algorithms

D.1 ZSL model

Due to the generality of the proposed framework, any ZSL model which learns a compatibility function can be used to fill this role. Reviewing the results as reported by (Xian et al. 2019), Attribute-Label Embedding (Akata et al. 2016) consistently achieves favourable per-class accuracy over other methods, ranking in top two for ZSL on the proposed split across the AWA2, CUB, SUN and APY datasets. Importantly, it also consistently achieves relatively high accuracy on the seen classes for the GZSL task. ALE learns a function $\theta(\boldsymbol{x})$ which embeds the image features into the same space as the class attributes, $\varphi(y)$. The compatibility function $F(\boldsymbol{x^{(i)}}, y)$ takes the form of the dot product between $\theta(\boldsymbol{x})$ and the $\varphi(y)$. An approximate ranking loss is used which penalises incorrect rankings of the compatibility score of the correct class label. The objective function is optimised through Stochastic Gradient Descent (SGD).

D.2 Novelty detection model

The model of choice for novelty detection is that of Bhattacharjee et al. (2020). The advantage of this model is that it requires no out-of-distribution data to be trained, unlike the model of (Perera & Patel 2019) which was another consideration. Furthermore, the former requires only

Table 2: TUNED HYPERPARAMETERS FOR ALE

Hyperparameter	AWA2	CUB	SUN	APY
# Epochs	15	80	40	60
Batch size	64	64	64	64
Weight decay	1e-4	1e-4	5e-4	1e-3
Learning rate	1e-4	1e-4	1e-4	1e-4

the features of the images which can be extracted from a pre-trained deep convolutional neural network such as ResNet (He et al. 2016), which is the exact same feature space used to train the ZSL model. This way, no extra data is required for this framework. Bhattacharjee et al. call their model the *mixing novelty detector*, which I will refer to hereon as MND. MND uses a 3-layer neural network which takes as input $(\boldsymbol{x}_i, \boldsymbol{x}_j, \boldsymbol{x}_k)$ where $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$ are two samples, $\boldsymbol{x}_k = \alpha\boldsymbol{x}_i + (1-\alpha)\boldsymbol{x}_j$ and $\alpha \in [0, 1]$ is a mixing coefficient. The model outputs a $|C|$-dimensional vector $\boldsymbol{u}$ where $\boldsymbol{u}_i$ estimates the weight of the ith class in the convex combination $\boldsymbol{x}_k$, and C is the set of training classes. The model is trained to optimise a constituency loss function

$$\mathcal{L}^{cons} = \sum_{r \in nm} \boldsymbol{u}_r^2 + g \sum_{r \in m} (\boldsymbol{u}_r - \boldsymbol{\beta}_r)^2 \tag{13}$$

where m denotes the mixing classes $\{y_i, y_j\}$, nm denotes the non-mixing classes $C \setminus \{y_i, y_j\}$, $\boldsymbol{\beta}$ is the ground-truth mixing coefficient vector s.t. the y_i-th entry is α and the y_j-th entry is $1 - \alpha$, and g is a hyperparameter. I slightly modify this loss function for my implementation to normalise it for the number of classes:

$$\mathcal{L}^{cons} = \frac{1}{|nm|} \sum_{r \in nm} \boldsymbol{u}_r^2 + \frac{g}{|m|} \sum_{r \in m} (\boldsymbol{u}_r - \boldsymbol{\beta}_r)^2. \tag{14}$$

This means that the optimal value of g should not change much, if at all, between datasets with different numbers of classes. Figure 4 supports this hypothesis, where the optimal novelty performance is achieved for $g \approx 0.5$. Like ALE, this loss is also minimised with SGD.

E Datasets

Animals with Attributes 2 (AWA2) (Xian et al. 2019) is a medium-scale course-grained dataset consisting of images of 50 species of animals labeled with 85 attributes. The authors provide both binary and continuous values for the attributes. The continuous values are used in this project. Caltech-UCSD-Birds 200-2011(CUB) (Welinder et al. 2010) is medium-scale fine-grained dataset of images of 200 species of birds labeled with 312 attributes. The Scene Understanding dataset (SUN) (Patterson & Hays 2012) is medium-scale and fine-grained, consisting of 717 types of scenes with 102 attributes. Attribute Pascal and Yahoo (APY) (Farhadi et al. 2009) is a small-scale course-grained dataset of images of various objects labeled with 64 attributes. Table 3 summarises the statistics of the four datasets. In their comprehensive evaluation of different ZSL models, Xian et al. proposed a new split for each of these datasets such that none of the unseen classes are present in ImageNet 1K, the dataset used to train the ResNet-101 model from which image features are extracted. This is important because feature extraction is part of the training process of ZSL, and indeed Xian et al. showed their proposed splits resulted in lower accuracy than the standard splits.

Table 3: DATASET SUMMARY STATISTICS

							Validation	
Dataset	Att	$\lvert\mathcal{Y}^s\rvert$	$\lvert\mathcal{Y}^u\rvert$	# Train	# Test	# Total	$\lvert\mathcal{Y}^s\rvert$	$\lvert\mathcal{Y}^u\rvert$
AWA2	85	40	10	23527	13795	37322	32	8
CUB	312	150	50	7057	4731	11788	112	38
SUN	102	645	72	10320	4020	14340	580	65
APY	64	20	12	5932	9407	15339	12	8

F Implementation

I implemented both ALE and MND in Python using the PyTorch library. The training and testing times were significantly reduced by running the tensor operations (forward and back propagation) on a GPU via Nvidia's CUDA (Compute Unified Device Architecture) API. All code was run on a laptop with an 8-core Intel i7-6700HQ @ 2.60GHz, a Quadro M1000M GPU and 16GB of memory. During the implementation process of the framework, one or two challenges arose which had to be dealt with in order for results to be obtained in a timely manner. For example, in the first iteration of the code, the novelty detection model (MND) and novelty function ν were trained and evaluated together in one program. This meant that obtaining results for a different ν required retraining the MND model needlessly. This was fixed by separating these processes into two programs with the intermediate results saved in a file, able to be reused by the different novelty functions.

Xian et al. have made the extracted ResNet-101 image features of each dataset that they used to evaluate models available to download from the MPI website. They also include their proposed split of each dataset as well as the normalised class attributes. This is where the datasets used in this project were sourced from. Hyperparameters were tuned across three random validation splits of the training data with the ratio of seen to unseen classes reflecting that of the whole dataset. Importantly, the novelty functions ν were only optimised on the validation splits of the training data, so they have knowledge of the test membership scores. The values of the parameters for ALE are shown in table 2. Figure 4 shows how changing the hyperparameter g in the constituency loss function (14) affects the performance of the novelty detection. This is measured as the area under curve (AUC) of the receiver operating characteristic (ROC), which measures the TPR (proportion of seen samples classified as seen) against the FPR (proportion of unseen classes classified as seen). This is the standard way of measuring the performance of any binary classifier.

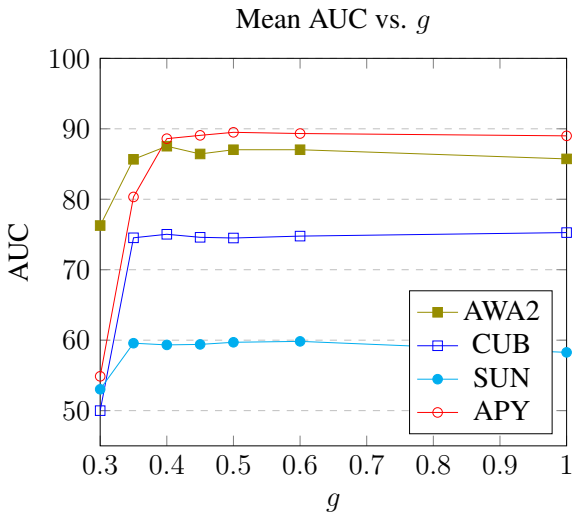

Figure 4: Effect of hyperparameter g on the novelty detection performance for each dataset. AUC is averaged over all 3 validation splits.

Table 4: GENERALIZED ZERO-SHOT LEARNING PER-CLASS TOP-1 ACCURACY

Method	Novelty function	AWA2			CUB			SUN			APY		
		u	s	H	u	s	H	u	s	H	u	s	H
DAP	–	0.0	84.7	0.0	1.7	67.9	3.3	4.2	25.1	7.2	4.8	78.3	9.0
IAP	–	0.9	87.6	1.8	0.2	72.8	0.4	1.0	37.8	1.8	5.7	65.6	10.4
DEVISE	–	17.1	74.7	27.8	23.8	53.0	32.8	16.9	27.4	20.9	4.9	76.9	9.2
SJE	–	8.0	73.9	14.4	23.5	59.2	33.6	14.7	30.5	19.8	3.7	55.7	6.9
SAE	–	1.1	82.2	2.2	7.8	54.0	13.6	8.8	18.0	11.8	0.4	80.9	0.9
ALE	–	13.7	90.1	23.9	25.8	67.7	37.3	24.9	38.9	30.3	18.6	73.1	29.7
CADA-VAE	–	55.8	75.0	63.9	51.6	53.5	52.4	47.2	35.7	40.6	–	–	–
MCGM-VAE	–	60.9	69.3	**64.8**	51.1	58.0	**54.3**	38.6	43.8	**41.1**	36.4	52.8	**43.1**
CMT	Yes	8.7	89.0	15.9	4.7	60.1	8.7	8.7	28.0	13.3	10.9	74.2	19.0
ALE	Autoencoder	–	–	–	40.1	40.0	40.1	–	–	–	35.3	53.0	42.4
SAE	Autoencoder	34.7	74.6	47.4	39.7	41.8	40.7	–	–	–	26.0	57.1	35.7
ALE+MND	$\nu_{threshold}$	46.8	77.3	58.3	29.1	60.9	39.4	44.6	23.5	30.8	27.8	73.1	40.2
	$\nu_{logistic}$	40.2	86.0	54.8	28.7	62.2	39.2	26.2	34.7	29.9	24.5	78.9	37.3
	$\nu_{logistic}$*	57.8	69.0	**62.9**	31.1	60.9	**41.2**	31.2	33.1	**32.1**	35.5	43.1	38.9
	ν_{cdf}	45.5	82.7	58.7	27.5	63.3	38.3	26.5	33.6	29.6	26.8	76.8	39.8
	ν_{cdf}*	54.3	69.4	60.9	28.3	62.4	38.9	27.6	33.1	30.1	33.4	59.4	42.8
	$\nu_{Gaussian}$	42.8	84.2	56.7	28.4	62.1	39.0	26.8	33.3	29.7	25.2	77.4	38.1
	$\nu_{Gaussian}$*	51.6	75.7	61.4	29.6	61.4	39.9	27.6	33.0	30.1	32.6	62.9	**43.0**

(* = de-biased, u = unseen classes, s = seen classes, H = harmonic mean of ts and tr). Highest H-score across traditional and novelty detection models shown in **bold.**

IV RESULTS

A Testing methodology

The novelty detection model (MND) was first trained on 3 random validation splits of the training data, with the ratio of seen to unseen classes reflecting that of the whole dataset. This ensures the novelty detection does not have any access to data for the unseen classes, giving it an unfair advantage. Each training epoch consists of splitting the training set into random pairs of samples and a choosing random mixing coefficients $\alpha \in [0, 1]$ for each pair, then minimising eq. (14). MND was trained for 20 epochs on AWA2 and APY and 25 epochs on CUB and SUN, using $g = 0.5$ for all. The novelty functions, ν, were optimised on the membership scores output by the model for the validation data. After this, the model was trained on the full training set. In parallel, the ZSL model (ALE) was trained on the full training set, the validation splits were only needed here to fine-tune the hyperparameters of the model. At test time, the membership scores of samples were calculated by MND, then ν was computed on these scores. Finally, the ZSL model used this information to classify the test sample.

For each dataset, the top-1 accuracy of the proposed framework is calculated for each class and then averaged across all, using the hyperparameters tuned on the validation sets. The accuracies on images of seen classes and unseen classes are calculated separately, and their harmonic mean is also taken, given by:

$$H = \frac{2 \times acc_u \times acc_s}{acc_u + acc_s}. \tag{15}$$

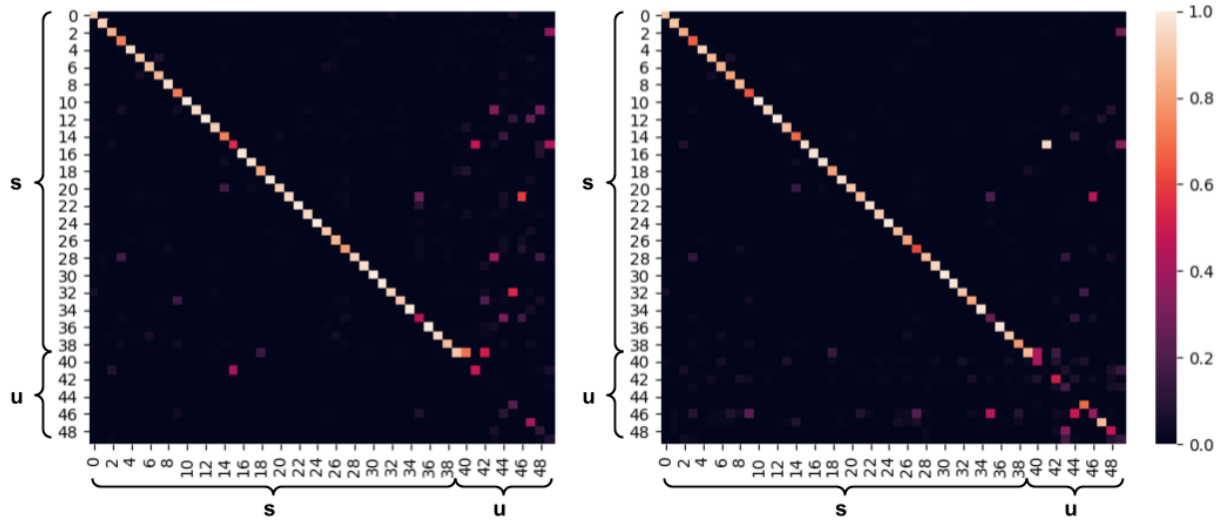

Figure 5: Confusion matrix visualisations for ALE *(left)* and ALE+MND with $\nu_{Gaussian}$ *(right)* on AWA2. Indices 0-39 represent seen classes and 40-49 the unseen classes. Novelty detection reduces the bias towards predicting seen classes.

This is the standard way to evaluate GZSL models since the work of Xian et al., and enables fair comparison to existing methods. The methods I compare to are five baseline methods, DAP and IAP (Lampert et al. 2014), DEVISE (Frome et al. 2013), SJE (Akata et al. 2015), and SAE (Kodirov et al. 2017); ALE (Akata et al. 2016) without novelty detection; two current state-of-the-art feature-generating models CADA-VAE (Schönfeld et al. 2019) and MCGM-VAE (Shao & Li 2020); CMT with outlier detection (Socher et al. 2013); and ALE and SAE with autoencoder-based novelty detection (Bhattacharjee et al. 2019). The results for DAP, IAP, DEVISE, SJE, SAE and CMT are taken from Xian et al., results for CADA-VAE, MCGM-VAE and ALE/SAE with autoencoder are taken from the original papers, and the results for ALE are from my own implementation. Table 4 shows the results.

B Analysis

The results show that the proposed approach outperforms the non-generative baseline methods and other existing novelty-detection-based approaches. Comparing the results of ALE to ALE+MND, the impact of using novelty detection can be seen. Using MND with $\nu_{logistic}$, the per-class accuracy on unseen classes increases by 60% on average across each dataset, and the accuracy on seen classes drops by 4%. When de-biasing is used with $\nu_{logistic}$, the test accuracy on unseen classes increases by 104%. However, the accuracy on seen classes drops by 22% on average. This suggests there is a trade-off between accuracy on seen and unseen classes on when de-biasing is used.

The confusion matrices for AWA2 in figure 5 show the the true label along the x-axis and the predicted label along the y-axis. It can be seen for ALE that unseen classes are usually incorrectly predicted as one of the seen classes. When novelty detection is used, there is a much stronger diagonal line and reduced spread for unseen classes, showing the increased accuracy and reduced bias.

Table 5: NOVELTY DETECTION PER-CLASS ACCURACY

Novelty function	**AWA2**		**CUB**		**SUN**		**APY**	
	u	s	u	s	u	s	u	s
$\nu_{threshold}$	77.8	79.2	56.1	75.0	74.7	43.0	82.3	74.5
$\nu_{logistic}$	73.2	83.2	55.7	75.4	48.5	69.4	77.1	78.8
ν_{cdf}	79.7	77.0	52.9	78.3	48.1	69.8	81.5	75.5
$\nu_{Gaussian}$	74.0	82.8	55.2	76.0	48.3	69.5	77.7	78.4

(u = unseen classes, s = seen classes)

V EVALUATION

A Strengths and weaknesses

It has been shown that using novelty detection to augment a ZSL model leads to significantly improved results in the generalised setting. The bias towards seen classes is reduced, thus the accuracy on unseen classes goes up. However, there are some clear limitations to using novelty detection techniques. Firstly, when simple thresholding is used, it is clear that the accuracy on unseen classes is limited by the sensitivity (TPR) of the novelty detector, while the accuracy on seen classes is limited by the specificity ($1 - \text{FPR}$). For the three probabilistic novelty functions, this limitation is not as hard ($\nu_{logistic}$ achieves 86.0% GZSL seen accuracy on AWA2 while having 83.2% novelty seen accuracy), but most of the time the GZSL accuracy is less than the novelty detection accuracy. A possible limitation of my approach is that it assumes the distribution of membership scores $N(\boldsymbol{x})$ of the validation set is close to that of the test set. This is truer for some datasets than others (see figure 3). In CUB, for example, the distribution of $N(\boldsymbol{x})$ for the unseen validation set is more positively skewed than that of the unseen test set.

Even though my framework outperforms the non-generative methods, the current feature-generating networks such as CADA-VAE and MCGM-VAE are the state of the art for the GZSL task. However, an advantage of the novelty detection approach is its flexibility, allowing the ZSL model and/or the novelty detection model to be swapped out for different choices (providing the ZSL model computes a compatibility function), as better models are developed. Moreover, with relatively small improvements to the novelty detection accuracy, the performance of my framework could overtake these generative approaches. One idea for how this could be achieved is by training the novelty detection model with features extracted from a CNN fine-tuned on the particular dataset being tested. In theory, this should lead to improved results on the fine-grained datasets such as CUB, where many of the different classes are grouped together as a single class in ImageNet 1K, meaning these generic features are not discriminative enough for the classification on the target dataset. Another idea for improving the accuracy could actually be to combine the generative approach with novelty detection such as was done by Manda et al. Synthesised image features could be used to train the novelty detection model without any modification–in the case of MND, the loss function (14) would enforce low output for all seen classes if the input were features of an unseen class.

B Project organisation

A lot of time early on in the project was spent reading the existing literature on zero-shot learning and gaining a deeper understanding of certain areas of machine learning in general. This was a crucial and unavoidable phase in the project, but already having this background knowledge would have allowed more time to be spent on developing the GZSL model. In assessing whether or not novelty detection was going to be a fruitful line of research to take the project in, more time was spent than should have been on implementing one of the novelty detection models in the literature (Perera & Patel 2019). This was due to the complexity of the model and time taken to train and test it (it involved fine-tuning AlexNet with the raw images on my laptop's relatively weak GPU). Eventually I ended up abandoning this model and looking for a simpler one (Bhattacharjee et al. 2020) which could be trained on pre-extracted image features and did not require any auxiliary dataset such as was required by Perera and Patel's model. Once I had found and read this paper, progress on the project increased rapidly and I found I as able to implement, test and make improvements to the framework quite quickly.

VI CONCLUSIONS

In this project, it has been investigated how a novelty detection model and a ZSL model can be combined in order to yield improved accuracy in the task of generalised zero-shot learning. The results show that the proposed approach significantly improves upon the baseline methods, other existing novelty-detection-based methods, and achieves near state-of-the-art performance on the AWA2 and APY datasets. A measure of a ZSL model's bias has been proposed, as well as a way to use this metric in the framework to give even higher accuracy, counteracting its effect. A variety of novelty functions, which convert the output of the novelty detection model to a probability, have been investigated, with logistic regression achieving the best results on three out of the four datasets when de-biasing is used.

Possible future lines of research include testing the novelty detection model with different ZSL models, investigating the possible improvements from using fine-tuned image features in the training process, and using the generated synthetic features for unseen classes to further improve the novelty detection accuracy. Two areas of ZSL neglected in this project were the transductive setting (where unlabeled data for unseen classes is available) and few-shot learning (a very low number of labeled unseen examples are available). The performance of this framework in these settings could also be investigated.

References

Akata, Z., Perronnin, F., Harchaoui, Z. & Schmid, C. (2016), 'Label-embedding for image classification', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **38**, 1425–1438.

Akata, Z., Reed, S., Walter, D., Lee, H. & Schiele, B. (2015), Evaluation of output embeddings for fine-grained image classification, *in* 'IEEE Conference on Computer Vision and Pattern Recognition', pp. 2927–2936.

Bhattacharjee, S., Mandal, D. & Biswas, S. (2019), Autoencoder based novelty detection for generalized zero shot learning, *in* '2019 IEEE International Conference on Image Processing', pp. 3646–3650.

Bhattacharjee, S., Mandal, D. & Biswas, S. (2020), Multi-class novelty detection using mix-up technique, *in* '2020 IEEE Winter Conference on Applications of Computer Vision'.

Chao, W., Changpinyo, S., Gong, B. & Sha, F. (2016), An empirical study and analysis of generalized zero-shot learning for object recognition in the wild, *in* 'The 14th European Conference on Computer Vision', Vol. 9906, pp. 52–68.

Farhadi, A., Endres, I., Hoiem, D. & Forsyth, D. (2009), Describing objects by their attributes, *in* '2009 IEEE Conference on Computer Vision and Pattern Recognition', pp. 1778–1785.

Frome, A., Corrado, G. S., Shlens, J., Bengio, S., Dean, J., Ranzato, M. A. & Mikolov, T. (2013), Devise: A deep visual-semantic embedding model, *in* 'Advances in Neural Information Processing Systems 26', Curran Associates, Inc., pp. 2121–2129.

Fu, Y., Hospedales, T., Xiang, T. & Gong, S. (2015), 'Transductive multi-view zero-shot learning', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **37**.

Goodfellow, I. J., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A. & Bengio, Y. (2014), Generative adversarial nets, *in* 'Proceedings of the 27th International Conference on Neural Information Processing Systems', Vol. 2 of *NIPS'14*, MIT Press, p. 2672–2680.

He, K., Zhang, X., Ren, S. & Sun, J. (2016), Deep residual learning for image recognition, *in* '2016 IEEE Conference on Computer Vision and Pattern Recognition', pp. 770–778.

Hu, J., Shen, L. & Sun, G. (2018), Squeeze-and-excitation networks, *in* 'Conference on Computer Vision and Pattern Recognition', pp. 7132–7141.

Kodirov, E., Xiang, T. & Gong, S. (2017), Semantic autoencoder for zero-shot learning, *in* 'IEEE Conference on Computer Vision and Pattern Recognition', pp. 4447–4456.

Lampert, C. H., Nickisch, H. & Harmeling, S. (2014), 'Attribute-based classification for zero-shot visual object categorization', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36**(3), 453–465.

Long, Y., Liu, L., Shao, L., Shen, F., Ding, G. & Han, J. (2017), From zero-shot learning to conventional supervised classification: Unseen visual data synthesis, *in* 'The IEEE Conference on Computer Vision and Pattern Recognition'.

Mandal, D., Narayan, S., Dwivedi, S., Gupta, V., Ahmed, S., Khan, F. S. & Shao, L. (2019), 'Out-of-distribution detection for generalized zero-shot action recognition', *Conference on Computer Vision and Pattern Recognition* pp. 9977–9985.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. & Dean, J. (2013), Distributed representations of words and phrases and their compositionality, *in* 'Proceedings of the 26th International Conference on Neural Information Processing Systems', Vol. 2, Curran Associates Inc., p. 3111–3119.

Patterson, G. & Hays, J. (2012), Sun attribute database: Discovering, annotating, and recognizing scene attributes, *in* '2012 IEEE Conference on Computer Vision and Pattern Recognition', pp. 2751–2758.

Perera, P. & Patel, V. M. (2019), Deep transfer learning for multiple class novelty detection, *in* '2019 IEEE Conference on Computer Vision and Pattern Recognition', pp. 11536–11544.

Schönfeld, E., Ebrahimi, S., Sinha, S., Darrell, T. & Akata, Z. (2019), Generalized zero- and few-shot learning via aligned variational autoencoders, *in* 'IEEE Conference on Computer Vision and Pattern Recognition', pp. 8239–8247.

Shao, J. & Li, X. (2020), 'Generalized zero-shot learning with multi-channel gaussian mixture vae', *IEEE Signal Processing Letters* **27**, 456–460.

Socher, R., Ganjoo, M., Bastani, H., Bastani, O., Manning, C. & Ng, A. (2013), 'Zero-shot learning through cross-modal transfer', *Advances in Neural Information Processing Systems* .

Welinder, P., Branson, S., Mita, T., Wah, C., Schroff, F., Belongie, S. & Perona, P. (2010), Caltech-ucsd birds 200, Technical Report CNS-TR-2010-001, California Institute of Technology.

Xian, Y., Lampert, C. H., Schiele, B. & Akata, Z. (2019), 'Zero-shot learning—a comprehensive evaluation of the good, the bad and the ugly', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **41**(9), 2251–2265.

Xian, Y., Lorenz, T., Schiele, B. & Akata, Z. (2018), Feature generating networks for zero-shot learning, *in* 'The IEEE Conference on Computer Vision and Pattern Recognition'.